Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of operating a computer system for evaluating a programming language statement that includes a first and a second sub-statement, the method comprising the steps of:

evaluating the first sub-statement and determining an evaluation success result if the evaluation succeeds or a distinguished value if evaluation fails; said distinguished value being a value not included in the range of possible evaluation success results of the first sub-statement;

determining whether the second sub-statement is to be evaluated, and if so, evaluating the second sub-statement and determining an evaluation success result if evaluation succeeds or said distinguished value if evaluation fails; a range of possible evaluation success results of the second sub-statement not including said distinguished value; and

determining an evaluation result of the statement depending on at least whether evaluation of the first sub-statement succeeds or fails;

wherein the evaluation success result of the first sub-statement and the evaluation success result of the second sub-statement belong to a first type in a set of types and the distinguished value belongs to a second type not in the set of types;

wherein each type in the set of types is not a supertype of the second type; and

wherein the first type in the set of types comprises one from the set of integer, float, Boolean, sequence, tuple, structure, multi-set, dictionary, string and enumeration.

2. (Original) The method of claim 1, wherein the second sub-statement is evaluated if evaluation of the first sub-statement did not fail, and the evaluation result of the statement is determined to be the evaluation success result of the second sub-statement if evaluation of the

first and the second sub-statements succeeds, and wherein the evaluation result of the statement is said distinguished value if evaluation of at least one of the first and second sub-statements fails.

3. (Original) The method of claim 1, wherein the second sub-statement is evaluated if evaluation of the first sub-statement fails, and wherein the evaluation result of the statement is the evaluation success result of the first sub-statement if evaluation of the first sub-statement succeeds; the evaluation result of the statement is the evaluation success result of the second sub-statement if evaluation of the first sub-statement fails but evaluation of the second sub-statement succeeds; and the evaluation result of the statement is said distinguished value if evaluation of both the first and the second sub-statements fails.

4. (Original) The method of claim 1, wherein the second sub-statement is evaluated concurrently with the evaluation of the first sub-statement, and the evaluation result of the statement is said distinguished value if evaluation of at least one of the first and second sub-statements fails.

5. (Original) The method of claim 1, wherein the second sub-statement is evaluated concurrently with the evaluation of the first sub-statement, and the evaluation result of the statement is said distinguished value only if evaluation of both the first and the second sub-statements fails.

6. (Original) The method of claim 1, wherein the second sub-statement is evaluated independently on whether evaluation of the first sub-statement succeeds, and the evaluation result of the statement is said distinguished value if evaluation of at least one of the first and second sub-statements fails.

7. (Original) The method of claim 1, wherein the second sub-statement is evaluated independently on whether evaluation of the first sub-statement succeeds, and the evaluation

result of the statement is said distinguished value if evaluation of both the first and second sub-statements fails.

8. (Original) The method of claim 1, wherein at least one of the first and second sub-statements includes a closure loop statement having an operand indicating that evaluation of the respective sub-statement does not stop before said operand evaluates to said distinguished value.

9. (Original) The method of claim 1, wherein at least one of the first or second sub-statements includes a rule statement having a first argument and a second argument, the evaluation of the first argument triggering the evaluation of the second argument.

10. (Original) The method of claim 1, wherein at least one of the first or second sub-statements includes an ordered action system.

11. (Original) The method of claim 1, wherein at least one of the first or second sub-statements includes an unordered action system.

12. (Original) The method of claim 1, wherein one of the first and second sub-statements is a declarative statement and the other one of the first and second sub-statements is an imperative statement.

13. (Original) The method of claim 1, wherein the first and second sub-statements are typed according to a hierarchy of types.

14. (Original) The method of claim 13, wherein said hierarchy of types includes at least one minimal type.

15. (Currently Amended) An article of manufacture for use in a computer system comprising:

a memory;

instructions stored in the memory for operating a method for evaluating a programming language statement that includes a first and a second sub-statement, the method comprising the steps of:

evaluating the first sub-statement and determining an evaluation success result if the evaluation succeeds or a distinguished value if evaluation fails; said distinguished value being a value not included in the range of possible evaluation success results of the first sub-statement;

determining whether the second sub-statement is to be evaluated, and if so, evaluating the second sub-statement and determining an evaluation success result if evaluation succeeds or said distinguished value if evaluation fails; a range of possible evaluation success results of the second sub-statement not including said distinguished value; and

determining an evaluation result of the statement depending on at least whether evaluation of the first sub-statement succeeds or fails;

wherein the evaluation success result of the first sub-statement and the evaluation success result of the second sub-statement belong to a first type in a set of types and the distinguished value belongs to a second type not in the set of types;

wherein the first type in the set of types is not a supertype of the second type; and

wherein each type in the set of types comprises one from the set of integer, float, Boolean, sequence, tuple, structure, multi-set, dictionary, string and enumeration.

16. (Currently Amended) A system for evaluating a programming language statement and determining an evaluation result of said statement; comprising:

a memory for storing the statement that includes a first and a second sub-statement, a processor for determining the evaluation result of the statement; the evaluation result of the statement depending on whether evaluation of the first and second sub-statements succeeds or fails; the processor being capable of evaluating the first sub-statement and determining an evaluation success result if evaluation succeeds, or a distinguished value if evaluation fails; the processor being capable of evaluating the second sub-statement and determining an evaluation success result if evaluation succeeds, or said distinguished value if evaluation fails; the evaluation success result of the first sub-statement and the evaluation success result of the second sub-statement belong to a first type in a set of types and the distinguished value belongs to a second type not in the set of types; wherein each type in the set of types is not a supertype of the second type; and wherein the first type in the set of types comprises one from the set of integer, float, Boolean, sequence, tuple, structure, multi-set, dictionary, string and enumeration.

17. (Canceled)

18. (Canceled)

19. (Previously Presented) The method of claim 1, wherein evaluation of at least one of the first sub-statement and the second sub-statement comprises a pattern matching operation.